Algorithms for reasoning with graphical models

Class 1
Rina Dechter

Dechter-Morgan&claypool book (Dbook): Chapters 1-2

Outline

- Graphical models: The constraint network, Probabilistic networks, cost networks and mixed networks. queries: consistency, counting, optimization and likelihood queries.
- Inference: Bucket elimination for deterministic networks (Adaptive-consistency, and the Davis-Putnam algorithms.) The induced-width
  - Inference: Bucket-elimination for Bayesian and Markov networks queries (mpe, map, marginal and probability of evidence)
  - Graph properties: induced-width, tree-width, chordal graphs, hypertrees, join-trees.
  - Inference: Tree-decomposition algorithms (join-tree propagation and junction-trees)
- Approximation by bounded Inference: (weighted Mini-bucket , belief/constraint-propagation, constraint propagation, generalized belief propagation, variational methods)
- Search for csps: Backtracking; pruning search by constraint propagation, backjumping and learning.
- Search: AND/OR search Spaces for likelihood, optimization queries (Probability of evidence, Partition function, MAP and MPE queries, AND/OR branch and bound).
- Approximation by sampling: Gibbs sampling, Importance sampling, cutset-sampling, SampleSearch and AND/OR sampling, Stochastic Local Search.
- Hybrid of search Inference: cutset-conditioning and cutset-sampling

Outline

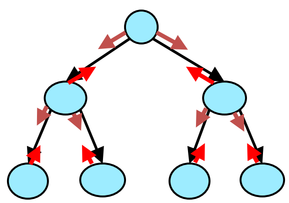
- Graphical models: The constraint network, Probabilistic networks, cost networks and mixed networks. Graphical representations and queries: consistency, counting, optimization and likelihood queries.
- Constraints inference: Bucket elimination for deterministic networks (Adaptive-consistency, and the Davis-Putnam algorithms.) The induced-width.
- Inference: Bucket-elimination for Bayesian and Markov networks queries (mpe,map, marginal and probability of evidence)
- Graph properties: induced-width, tree-width, chordal graphs, hypertrees, join-trees.
- Inference: Tree-decomposition algorithms (join-tree propagation and junction-trees algorithm, Cluster tree-elimination.)
- Approximation by bounded Inference: (Mini-bucket , belief-propagation, constraint propagation, generalized belief propagation)
- Search: Backtracking search algorithms; pruning search by constraint propagation, backjumping and learning.

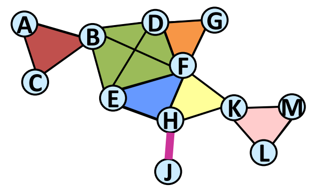
Course Requirements/Textbook

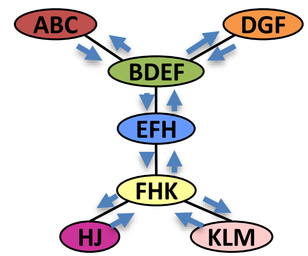
- Homeworks : There will be 5-6 problem sets , graded 70% of the final grades.
- A term project: paper presentation, a programming project.
- Books:
  - "Reasoning with probabilistic and deterministic graphical models", R. Dechter, Claypool, 2013 https://www.morganclaypool.com/doi/abs/10.2200/S00529ED1V01Y201308AIM023
  - "Modeling and Reasoning with Bayesian Networks", A. Darwiche, MIT Press, 2009.
  - "Constraint Processing" , R. Dechter, Morgan Kauffman, 2003

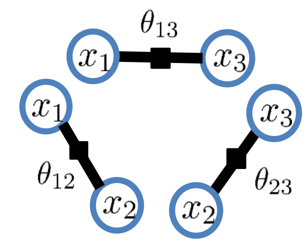
Outline of classes

- Part 1: Introduction and Inference

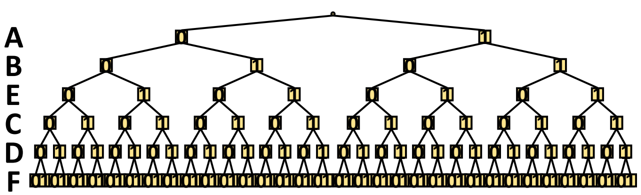


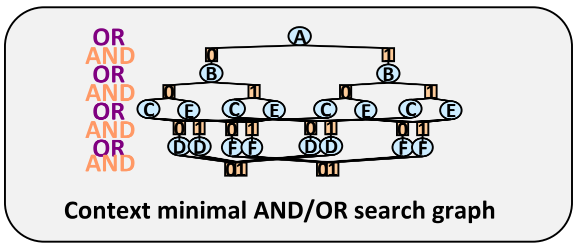




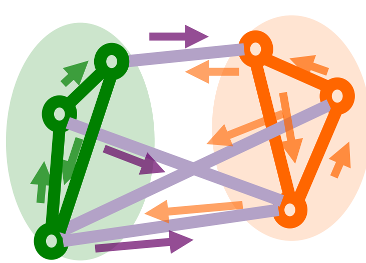


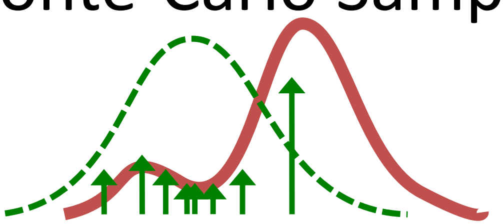
- Part 2: Search





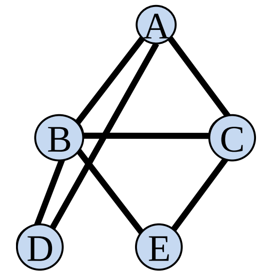
- Parr 3: Variational Methods and Monte-Carlo Sampling



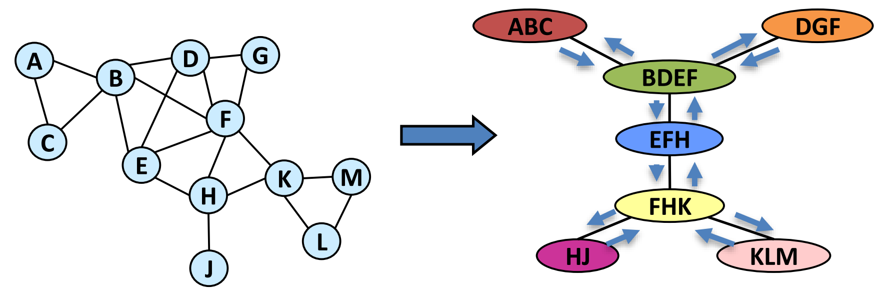


RoadMap: Introduction and Inference

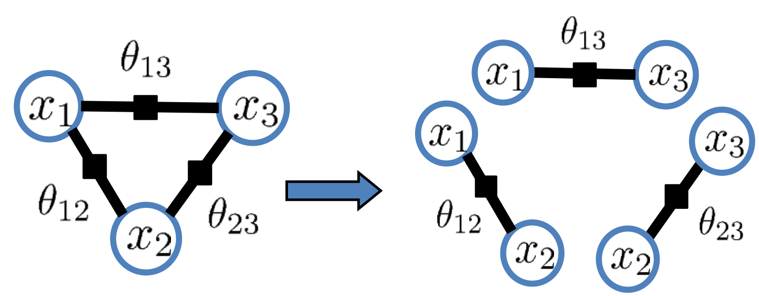
- Basics of graphical models
  - Queries
  - Examples, applications, and tasks
  - Algorithms overview



- Inference algorithms, exact
  - Bucket elimination for trees
  - Bucket elimination
  - Jointree clustering
  - Elimination orders



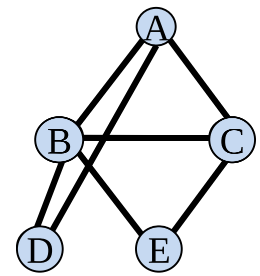
- Approximate elimination
  - Decomposition bounds
  - Mini-bucket & weighted mini-bucket
  - Belief propagation



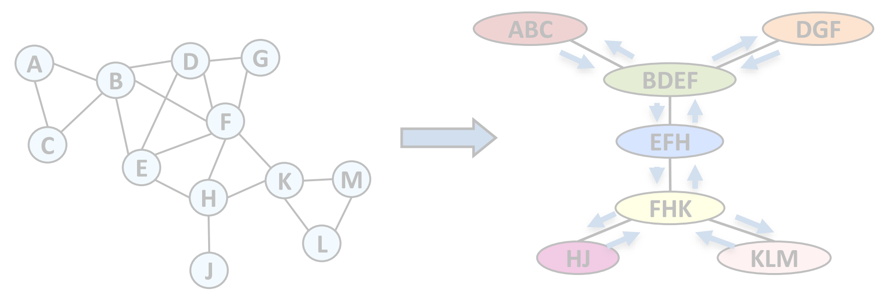
- Summary and Part 2

RoadMap: Introduction and Inference

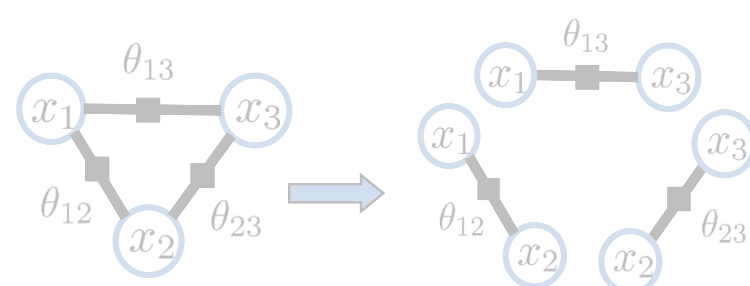
- Basics of graphical models
  - Queries
  - Examples, applications, and tasks
  - Algorithms overview



- Inference algorithms, exact
  - Bucket elimination for trees
  - Bucket elimination
  - Jointree clustering
  - Elimination orders



- Approximate elimination
  - Decomposition bounds
  - Mini-bucket & weighted mini-bucket
  - Belief propagation



- Summary and Class 2

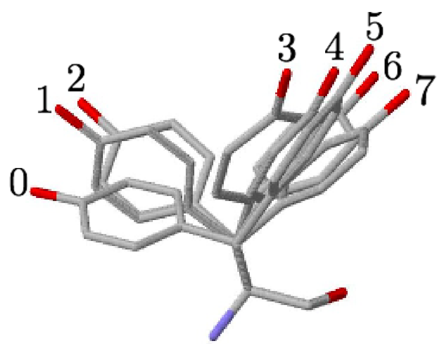
Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence

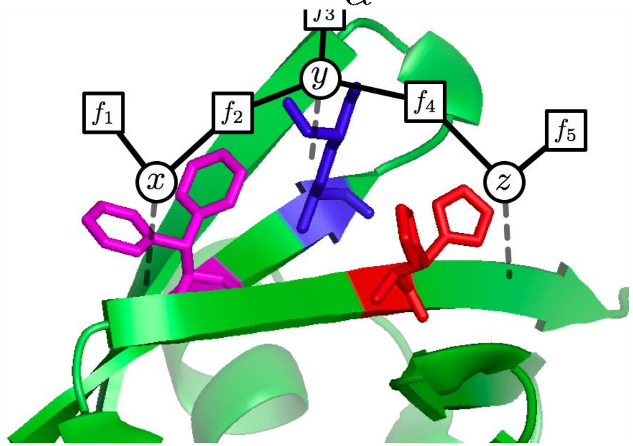
Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of "smaller", "local" interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Maximization (MAP): compute the most probable configuration

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \prod_\alpha f_\alpha(\mathbf{x}_\alpha) \qquad f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$



Phenylalanine



[Yanover & Weiss 2002]

Probabilistic Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of “smaller”, “local” interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Summation & marginalization

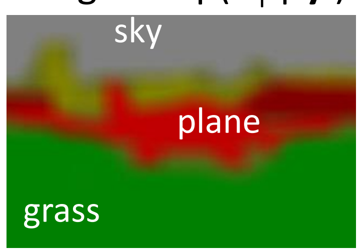
“partition function”

$$p(x_i) = \frac{1}{Z} \sum_{\mathbf{x} \setminus x_i} \prod_\alpha f_\alpha(\mathbf{x}_\alpha) \quad \text{and} \quad Z = \sum_{\mathbf{x}} \prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$

Observation **y**

$\Rightarrow$

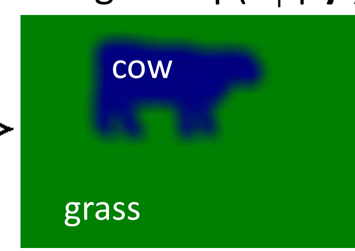
Marginals p(x_i | **y**)



Observation **y**

$\Rightarrow$

Marginals p(x_i | **y**)



e.g., [Plath et al. 2009]

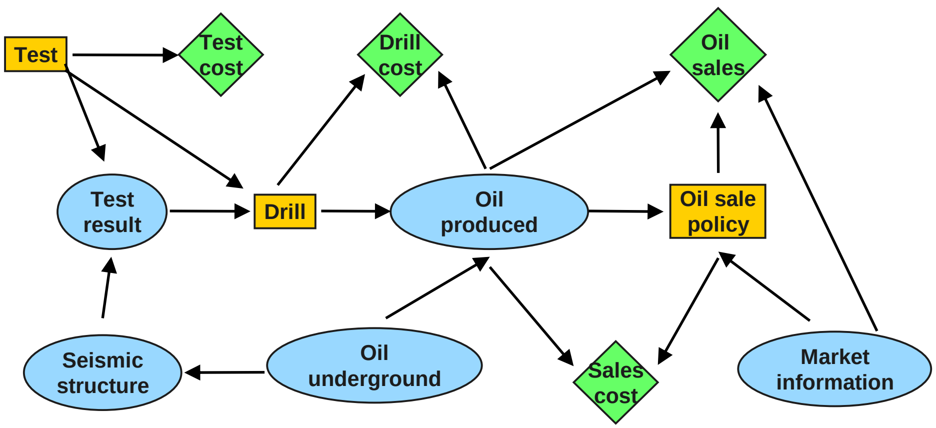
Graphical models

- Describe structure in large problems
  - Large complex system $F(X)$
  - Made of "smaller", "local" interactions $f_\alpha(x_\alpha)$
  - Complexity emerges through interdependence
- Examples & Tasks
  - Mixed inference (marginal MAP, MEU, …)

$$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_\alpha f_\alpha(\mathbf{x}_\alpha)$$

Influence diagrams & optimal decision-making

(the "oil wildcatter" problem)



e.g., [Raiffa 1968; Shachter 1986]

In more details…

Constraint Networks

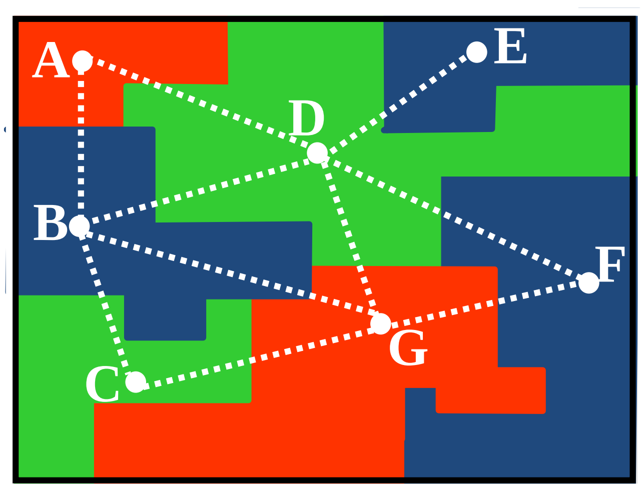
Example: map coloring

Variables - countries (A,B,C,etc.)

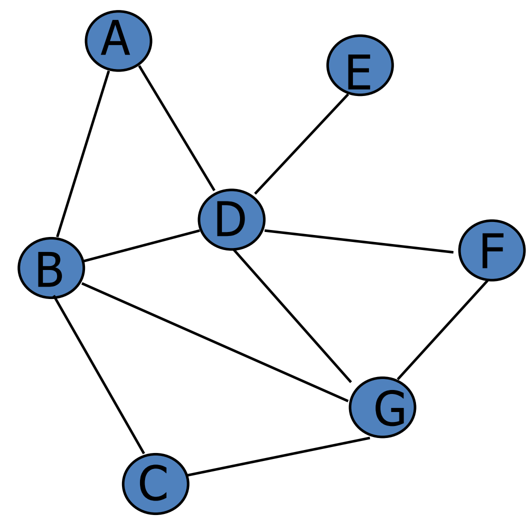
Values - colors (red, green, blue)

Constraints: $A \neq B$, $A \neq D$, $D \neq E$, *etc.*

A	B
red	green
red	yellow
green	red
green	yellow
yellow	green
yellow	red



Constraint graph

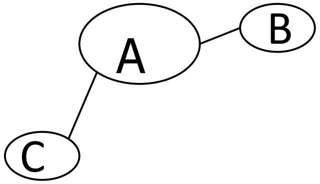


Propositional Reasoning

Example: party problem

- If $\overset{=A}{\boxed{\text{Alex goes}}}$, then $\overset{=B}{\boxed{\text{Becky goes}}}$: $\quad$ $\mathbf{A} \rightarrow \mathbf{B}$
- If $\underset{=C}{\boxed{\text{Chris goes}}}$, then $\underset{=A}{\boxed{\text{Alex goes}}}$: $\quad$ $\mathbf{C} \rightarrow \mathbf{A}$

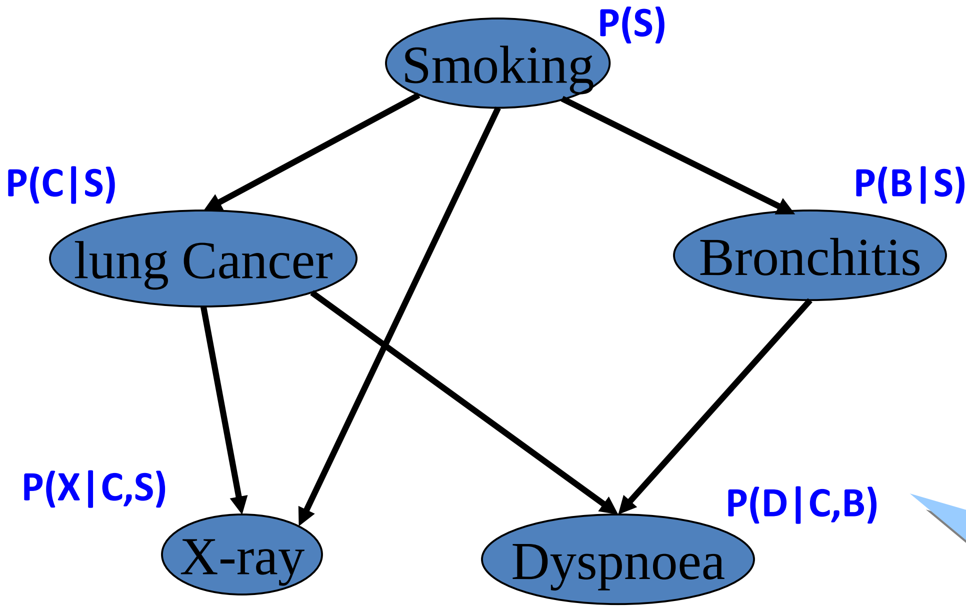
- **Question:**
  Is it possible that Chris goes to the party but Becky does not?



Is the *propositional theory*

$$\varphi = \{A \rightarrow B, C \rightarrow A, \neg \mathbf{B}, \mathbf{C}\}$$ satisfiable?

Bayesian Networks (Pearl 1988)



BN = (G, Θ)

CPD:

C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

P(S, C, B, X, D) = $P(S)\,P(C|S)\,P(B|S)\,P(X|C,S)\,P(D|C,B)$

Combination: Product
Marginalization: sum/max

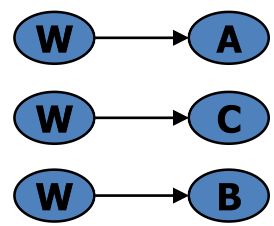
- Posterior marginals, probability of evidence, MPE
- P(D= 0) = $\sum_{S,L,B,X}$ P(S)· P(C|S)· P(B|S)· P(X|C,S)· P(D|C,B

MAP(P)= $max_{S,L,B,X}$ P(S)· P(C|S)· P(B|S)· P(X|C,S)· P(D|C,B)

Probabilistic reasoning (directed)

Party example: the weather effect

- Alex is-likely-to-go in bad weather
- Chris rarely-goes in bad weather
- Becky is indifferent but unpredictable



P(A|W=bad)=.9

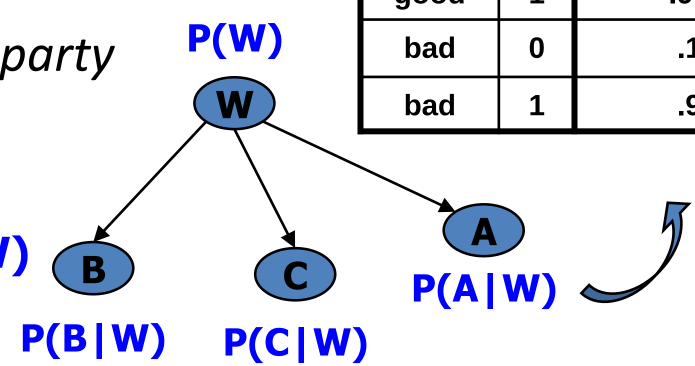
P(C|W=bad)=.1

P(B|W=bad)=.5

Questions:

- *Given bad weather, which group of individuals is most likely to show up at the party?*
- *What is the probability that Chris goes to the party but Becky does not?*

W	A	P(A|W)
good	0	.01
good	1	.99
bad	0	.1
bad	1	.9



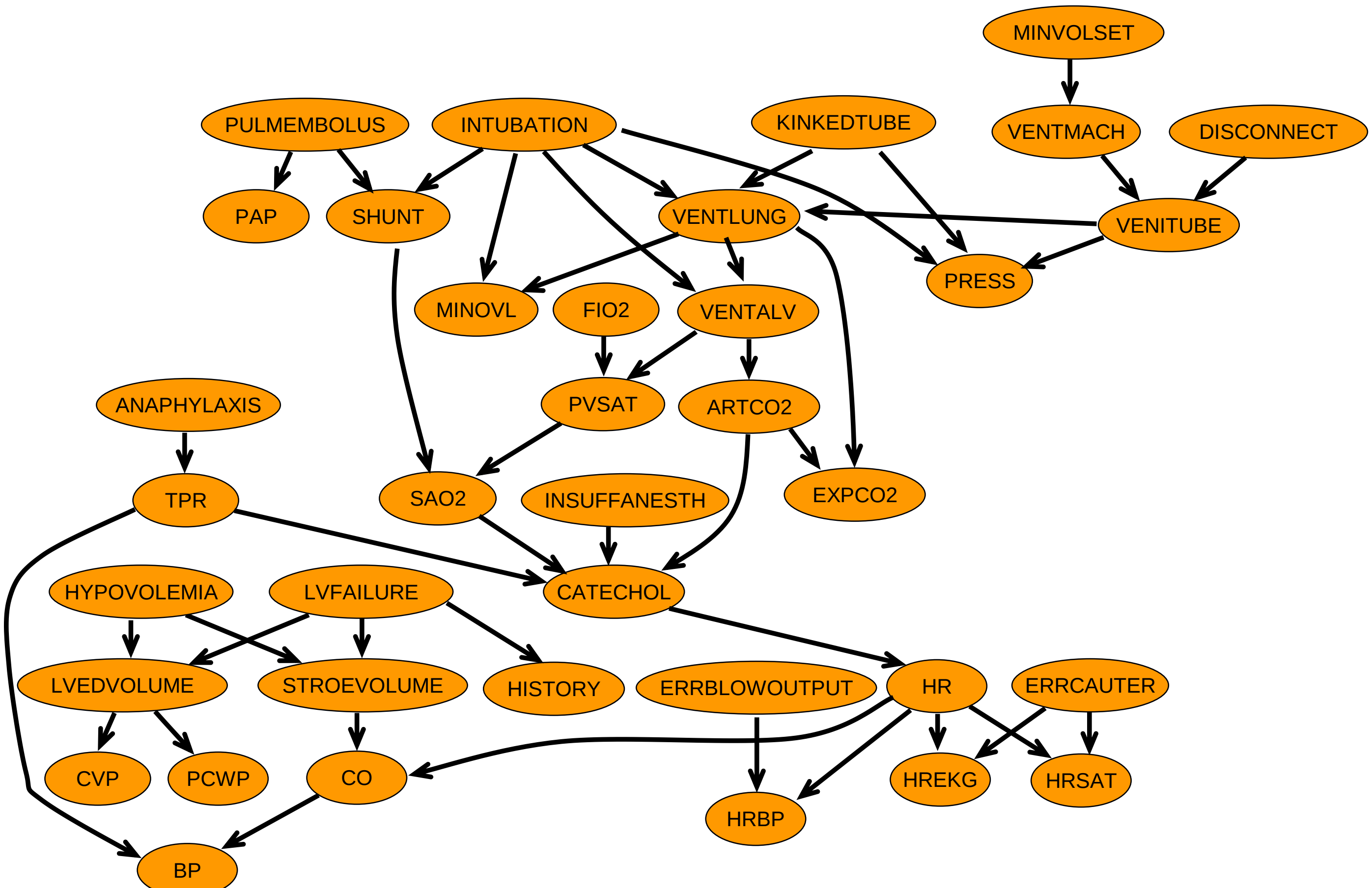
$$P(W,A,C,B) = P(B|W) \cdot P(C|W) \cdot P(A|W) \cdot P(W)$$

$$P(A,C,B|W=bad) = 0.9 \cdot 0.1 \cdot 0.5$$

Alarm network [Beinlich et al., 1989]

- Bayes nets: compact representation of large joint distributions

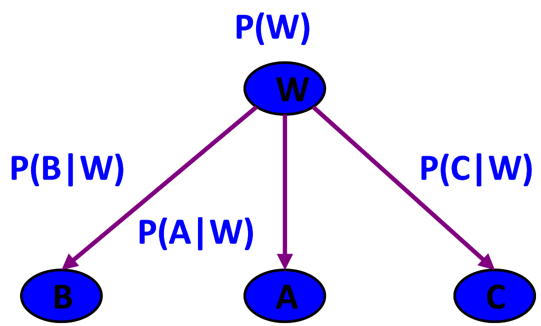
The “alarm” network: 37 variables, 509 parameters (rather than $2^{37} = 10^{11}$!)



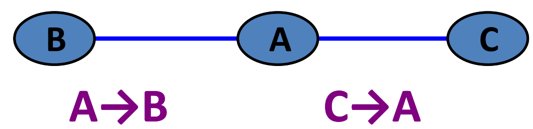
Mixed Probabilistic and Deterministic networks

Alex is-likely-to-go in bad weather
Chris rarely-goes in bad weather
Becky is indifferent but unpredictable

PN



CN



Query:
Is it likely that Chris goes to the party if Becky does not but the weather is bad?

$$P(C, \neg B \mid w = bad, A \to B, C \to A)$$

Graphical models (cost networks)

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Example:

$A \in \{0,1\}$

$B \in \{0,1\}$

$C \in \{0,1\}$

$f_{AB}(A,B), \quad f_{BC}(B,C)$

The *combination operator* defines an overall function from the individual factors,

e.g., "+" : $F(A,B,C) = f_{AB}(A,B) + f_{BC}(B,C)$

Notation:

Discrete Xi values called states

Tuple or configuration: states taken by a set of variables

Scope of f: set of variables that are arguments to a factor f

often index factors by their scope, e.g., $f_\alpha(X_\alpha), \quad X_\alpha \subseteq X$

Graphical models (cost networks)

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains (we'll assume discrete)

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

and a *combination operator*

Example:

$A \in \{0, 1\}$

$B \in \{0, 1\}$

$C \in \{0, 1\}$

$f_{AB}(A, B), \quad f_{BC}(B, C)$

$$F(A, B, C) = f_{AB}(A, B) + f_{BC}(B, C)$$

For discrete variables, think of functions as "tables"
(though we might represent them more efficiently)

A	B	f(A,B)
0	0	6
0	1	0
1	0	0
1	1	6

+

B	C	f(B,C)
0	0	6
0	1	0
1	0	0
1	1	6

=

A	B	C	f(A,B,C)
0	0	0	12
0	0	1	6
0	1	0	0
0	1	1	6
1	0	0	6
1	0	1	0
1	1	0	6
1	1	1	12

= 0 + 6

$F(A = 0, B = 1, C = 1)$

Graph Visualiization: Primal Graph

A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

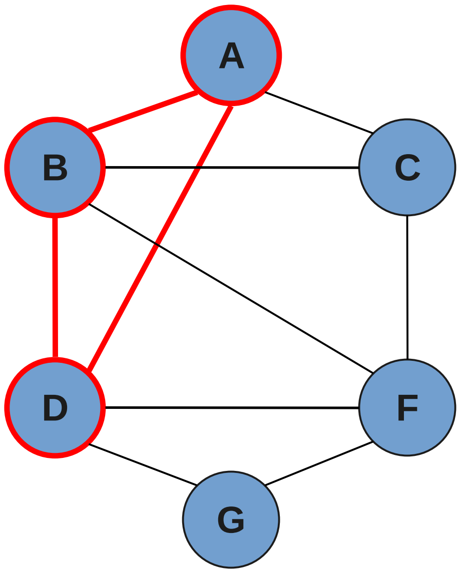
and a *combination operator*

Primal graph:

variables → nodes

factors → cliques

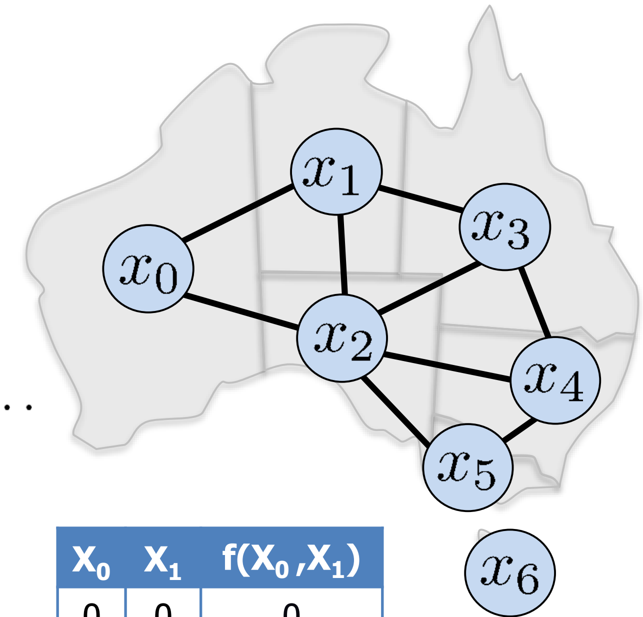
$$F(A,B,C,D,F,G) = f_1(A,B,D) + f_2(D,F,G) + f_3(B,C,F) + f_4(A,C)$$



Example: Constraint networks

$X_i \in \{red, green, blue\}$

$f_{ij}(X_i, X_j) = (X_i \neq X_j)$ for adjacent regions i,j



Overall function is "and" of individual constraints:

$$F(X) = f_{01}(X_0, X_1) \wedge f_{12}(X_1, X_2) \wedge f_{02}(X_0, X_2) \wedge \ldots$$

"Tabular" form:

$$f_{ij}(X_i, X_j) = \begin{cases} 1.0 & X_i \neq X_j \\ 0.0 & X_i = X_j \end{cases}$$

$$F(X) = \prod_{ij} f_{ij}(X_i, X_j) = \begin{cases} 1.0 & \text{all valid} \\ 0.0 & \text{any invalid} \end{cases}$$

X_0	X_1	$f(X_0, X_1)$
0	0	0
0	1	1
0	2	1
1	0	1
1	1	0
1	2	1
2	0	1
2	1	1
2	2	0

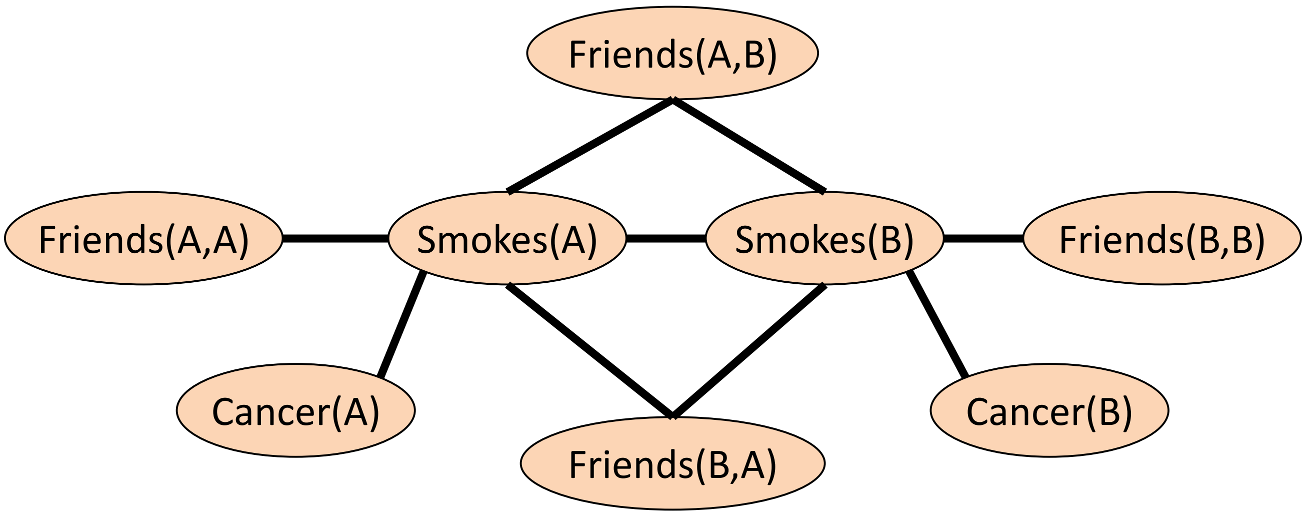
Tasks: "max": is there a solution?
"sum": how many solutions?

Markov logic, Markov networks

[Richardson & Domingos 2005]

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)



S_A	C_A	$f(S_A,C_A)$
0	0	exp(1.5)
0	1	exp(1.5)
1	0	1.0
1	1	exp(1.5)

F_{AB}	S_A	S_B	f(.)
0	0	0	exp(1.1)
0	0	1	exp(1.1)
0	1	0	exp(1.1)
0	1	1	exp(1.1)
1	0	0	exp(1.1)
1	0	1	1.0
1	1	0	1.0
1	1	1	exp(1.1)

Graphical visualization

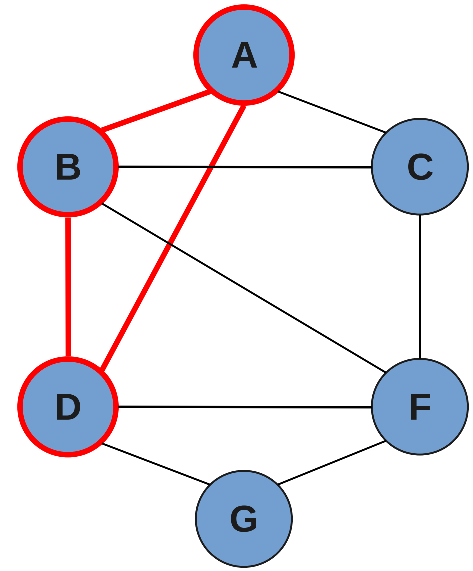
A *graphical model* consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or “factors”

and a *combination operator*

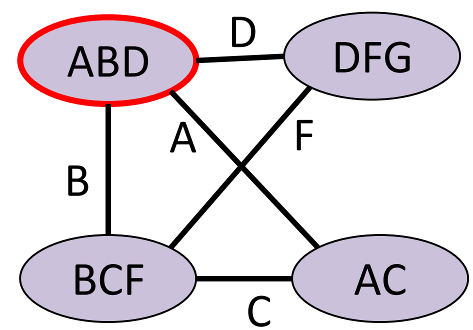


Primal graph:

variables $\longrightarrow$ nodes

factors $\longrightarrow$ cliques

$$F(A,B,C,D,F,G) = f_1(A,B,D) + f_2(D,F,G) + f_3(B,C,F) + f_4(A,C)$$



Dual graph:

factor scopes $\longrightarrow$ nodes

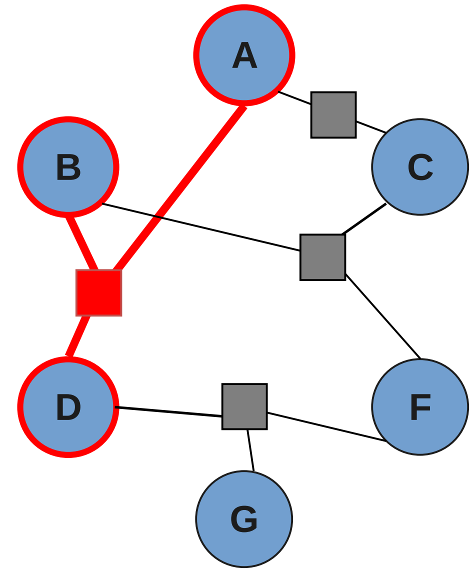
edges $\longrightarrow$ intersections (separators)

Graphical visualization

"Factor" graph: explicitly indicate the scope of each factor

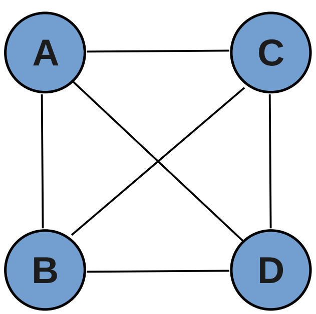
variables $\rightarrow$ circles

factors $\rightarrow$ squares



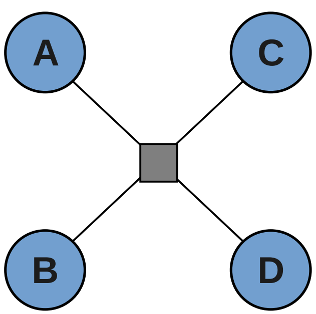
$$F(A,B,C,D,F,G) = f_1(A,B,D) + f_2(D,F,G) + f_3(B,C,F) + f_4(A,C)$$

Useful for disambiguating factorization:



$\stackrel{?}{=}$

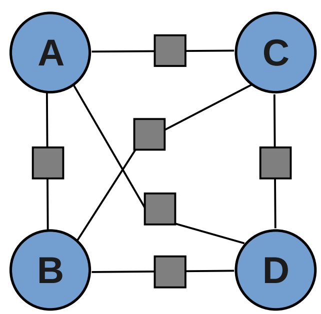
$f_1(A,B,C,D)$



$O(d^4)$

vs.

$f_1(A,B) + f_2(A,C) + \ldots$



pairwise: $O(d^2)$

Graphical models

A ***graphical model*** consists of:

$X = \{X_1, \ldots, X_n\}$ -- variables

$D = \{D_1, \ldots, D_n\}$ -- domains

$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ -- functions or "factors"

Operators:

combination operator

(sum, product, join, ...)

elimination operator

(projection, sum, max, min, ...)

Types of queries:

Marginal: $Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

MPE / MAP: $f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

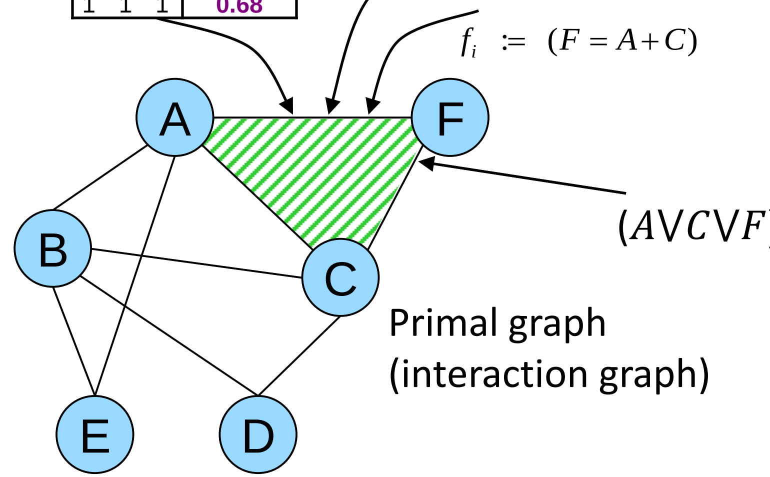
Marginal MAP: $f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

Conditional Probability Table (CPT)

A	C	F	P(F|A,C)
0	0	0	0.14
0	0	1	0.96
0	1	0	0.40
0	1	1	0.60
1	0	0	0.35
1	0	1	0.65
1	1	0	0.72
1	1	1	0.68

Relation

A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue



- All these tasks are NP-hard
  - exploit problem structure
  - identify special cases
  - approximate

Graphical models/reasoning task

Definition 2.1.2 (graphical model) *A* graphical model $\mathcal{M}$ *is a 4-tuple,* $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \otimes \rangle$, *where:*

1. $\mathbf{X} = \{X_1, \ldots, X_n\}$ *is a finite set of variables;*
2. $\mathbf{D} = \{D_1, \ldots, D_n\}$ *is the set of their respective finite domains of values;*
3. $\mathbf{F} = \{f_1, \ldots, f_r\}$ *is a set of positive real-valued discrete functions, defined scopes of variables* $\mathbf{S}_i \subseteq \mathbf{X}$,
4. $\otimes$ *is a combination operator*[1] *(e.g.,* $\otimes \in \{\prod, \sum, \bowtie\}$ *(product, sum, join)).*

The graphical model represents a global function *whose scope is* $\mathbf{X}$ *which is the combination of all its functions:* $\otimes_{i=1}^{r} f_i$.

Definition 2.1.3 (a reasoning problem) *A* reasoning problem *over a graphical model* $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \otimes \rangle$ *and a subset of variable* $Y \subset \mathbf{X}$ *is defined by a marginalization operator* $\Downarrow_{\mathbf{Y}}$. *If* $\mathbf{S}$ *is the scope of function* f *then* $\Downarrow_{\mathbf{Y}} f \in \{\max_{\mathbf{S}-\mathbf{Y}} f, \min_{\mathbf{S}-\mathbf{Y}} f, \pi_{\mathbf{Y}} f, \sum_{\mathbf{S}-\mathbf{Y}} f\}$ *is a marginalization operator. The reasoning problem* $\mathcal{P}\langle \mathcal{M}, \Downarrow_{\mathbf{Y}}, \mathbf{Z} \rangle$ *is the task of computing the function* $\mathcal{P}_{\mathcal{M}}(\mathbf{Z}) = \Downarrow_{\mathbf{Z}} \otimes_{i=1}^{r} f_i$, *where* r *is the number of functions in* F.

Summary of graphical models types

- Constraint networks
- Cost networks
- Bayesian network
- Markov networks
- Mixed probability and constraint network
- Influence diagrams

Constraint Networks

Map coloring

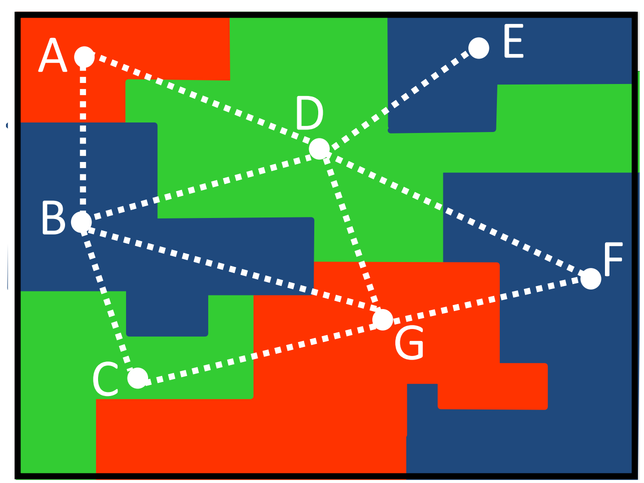
Combination = join
Marginalization = projection

Variables: countries (A B C etc.)

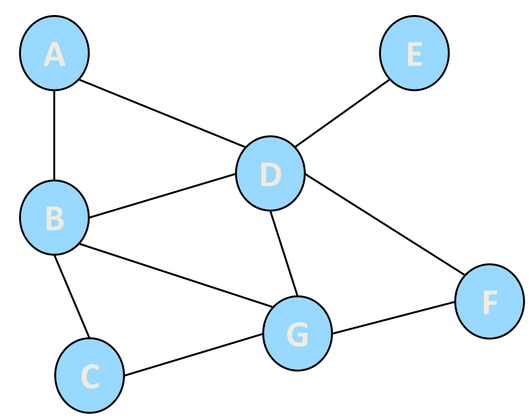
Values: colors (red green blue)

Constraints: $\mathbf{A \neq B}$, $\mathbf{A \neq D}$, $\mathbf{D \neq E}$, …

A	B
red	green
red	yellow
green	red
green	yellow
yellow	green
yellow	red



Constraint graph



Queries: Find one solution, all solutions, counting

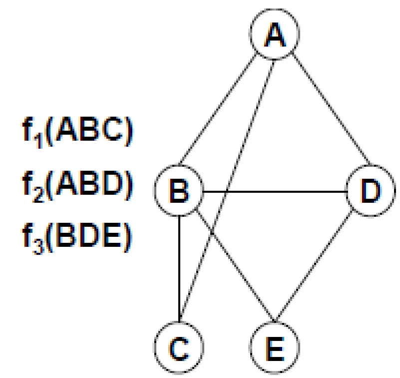
Example of a Cost Network

A	B	C	$f_1(ABC)$
0	0	0	∞
0	0	1	∞
0	1	0	∞
0	1	1	2
1	0	0	∞
1	0	1	2
1	1	0	∞
1	1	1	2

A	B	D	$f_2(ABD)$
0	0	0	1
0	0	1	∞
0	1	0	0
0	1	1	2
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

B	D	E	$f_3(BDE)$
0	0	0	∞
0	0	1	3
0	1	0	∞
0	1	1	4
1	0	0	∞
1	0	1	3
1	1	0	∞
1	1	1	4

(a) Cost functions



(b) Constraint graph

Figure 2.3: A cost network.

Combination: sum
Marginalization:min/max

Definition 2.3.2 (WCSP) *A Weighted Constraint Satisfaction Problem (WCSP) is a graphical model* $\langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \sum \rangle$ *where each of the functions* $f_i \in \mathbf{F}$ *assigns "0" (no penalty) to allowed tuples and a positive integer penalty cost to the forbidden tuples. Namely,* $f_i : D_{X_{i_1}} \times ... \times D_{X_{i_t}} \rightarrow \mathbb{N}$*, where* $S_i = \{X_{i_1}, ..., X_{i_t}\}$ *is the scope of the function.*

A Bayesian Network

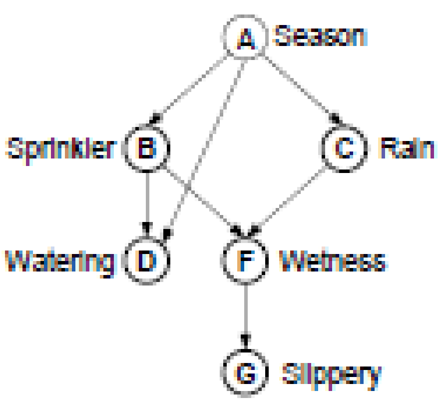
B	C	F	$P(F\|B,C)$
false	false	true	0.1
true	false	true	0.9
false	true	true	0.8
true	true	true	0.95

B	$A = winter$	D	$P(D\|A,B)$
false	false	true	0.3
true	false	true	0.9
false	true	true	0.1
true	true	true	1

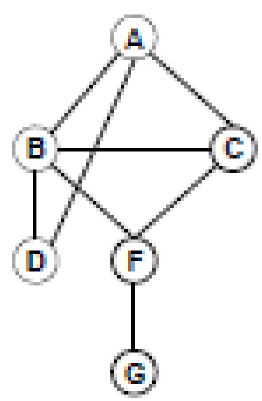
A	C	$P(C\|A)$
Summer	true	0.1
Fall	true	0.4
Winter	true	0.9
Spring	true	0.3

A	B	$P(B\|A)$
Summer	true	0.8
Fall	true	0.4
Winter	true	0.1
Spring	true	0.6

F	G	$P(G\|F)$
false	true	0.1
true	true	1



(a) Directed acyclic graph

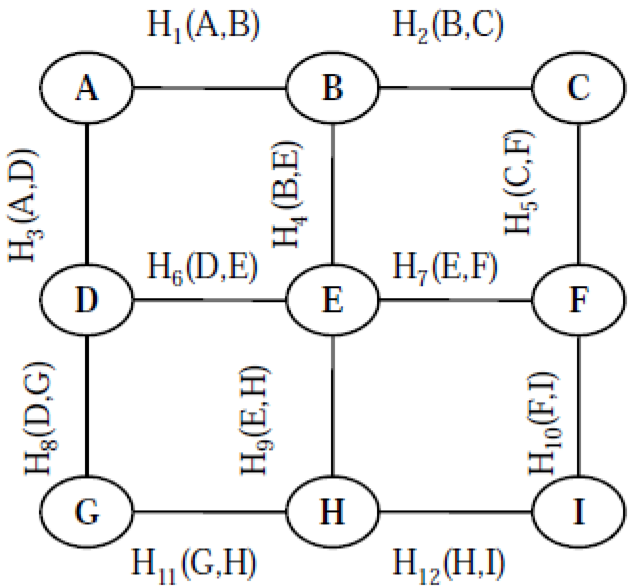


(b) Moral graph

Combination: product
Marginalization: sum or min/max

Belief network $P(g, f, c, b, a) = P(g|f)P(f|c, b)P(d|a, b)P(c|1)P(b|a)P(a)$

Markov Networks



(a)

D	E	H_6(D,E)
0	0	20.2
0	1	12
1	0	23.4
1	1	11.7

(b)

Figure 2.6: (a) An example 3×3 square Grid Markov network (ising model) and (b) An example potential $H_6(D, E)$

network represents a global joint distribution over the variables $\boldsymbol{X}$ *given by:*

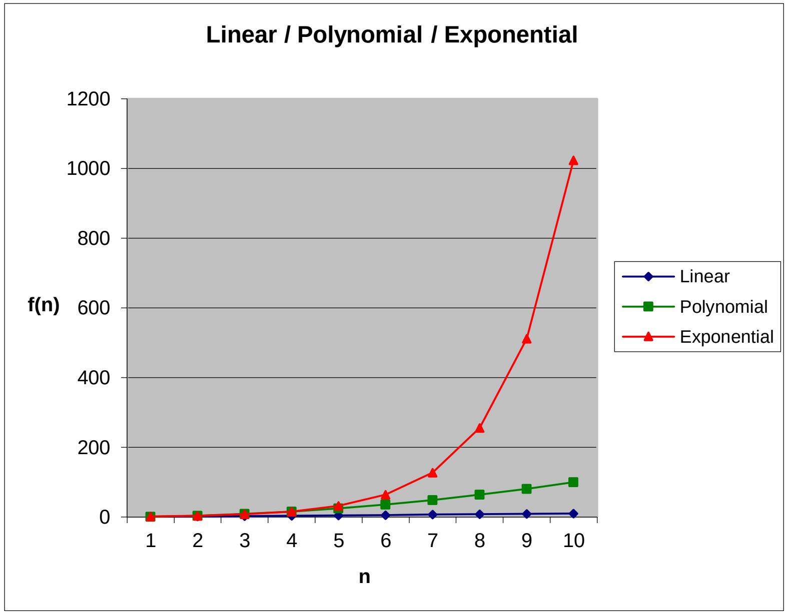
$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{i=1}^{m} H_i(\boldsymbol{x}) \quad , \quad Z = \sum_{\boldsymbol{x} \in \boldsymbol{X}} \prod_{i=1}^{m} H_i(\boldsymbol{x})$$

Example domains for graphical models

- Natural Language processing
  - Information extraction, semantic parsing, translation, topic models, …
- Computer vision
  - Object recognition, scene analysis, segmentation, tracking, …
- Computational biology
  - Pedigree analysis, protein folding and binding, sequence matching, …
- Networks
  - Webpage link analysis, social networks, communications, citations, ….
- Robotics
  - Planning & decision making

Complexity of Reasoning Tasks

- Constraint satisfaction
- Counting solutions
- Combinatorial optimization
- Belief updating
- Most probable explanation
- Decision-theoretic planning

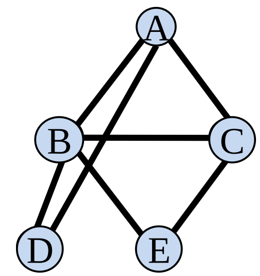


Reasoning is computationally hard

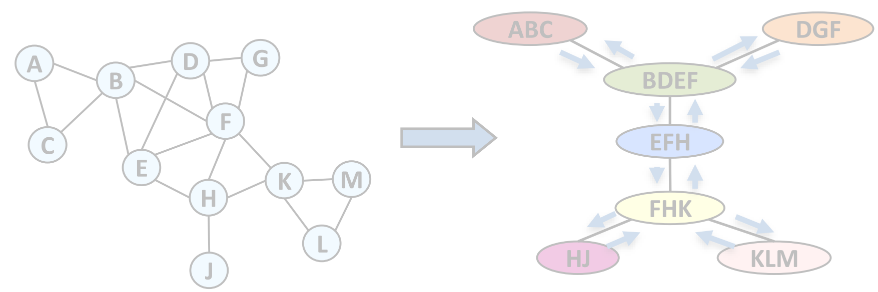
Complexity is
Time and space(memory)

RoadMap: Introduction and Inference

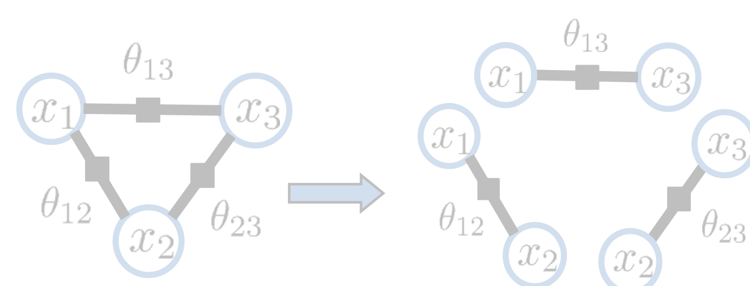
- Basics of graphical models
  - Queries
  - Examples, applications, and tasks
  - **Algorithms overview**



- Inference algorithms, exact
  - Bucket elimination for trees
  - Bucket elimination
  - Jointree clustering
  - Elimination orders



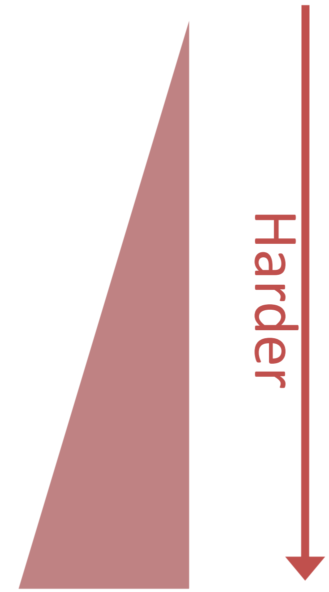
- Approximate elimination
  - Decomposition bounds
  - Mini-bucket & weighted mini-bucket
  - Belief propagation



- Summary and Class 2

Types of queries

Max-Inference	$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Sum-Inference	$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Mixed-Inference	$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

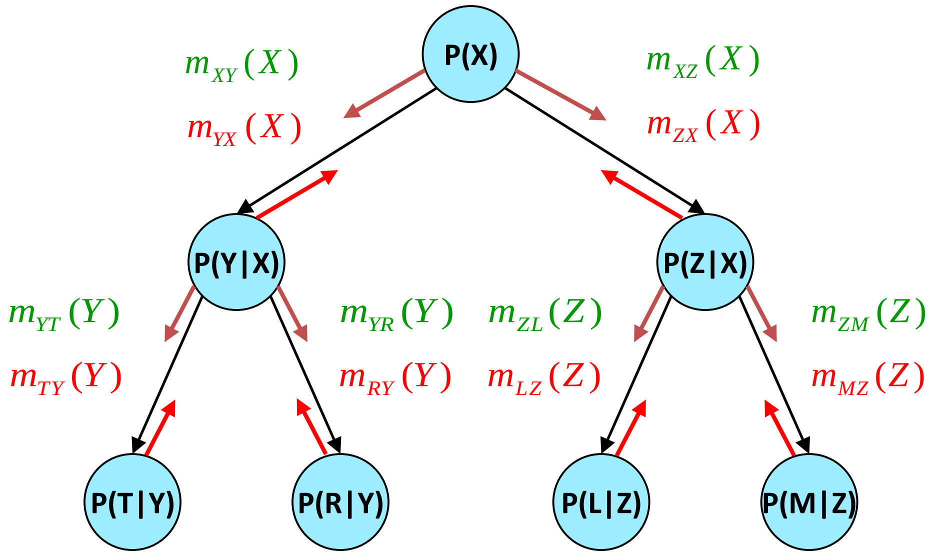


- **NP-hard**: exponentially many terms
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate ! Slower & more accurate

Tree-solving is easy

CSP – consistency (projection-join)

Belief updating (sum-prod)



MPE (max-prod)

#CSP (sum-prod)

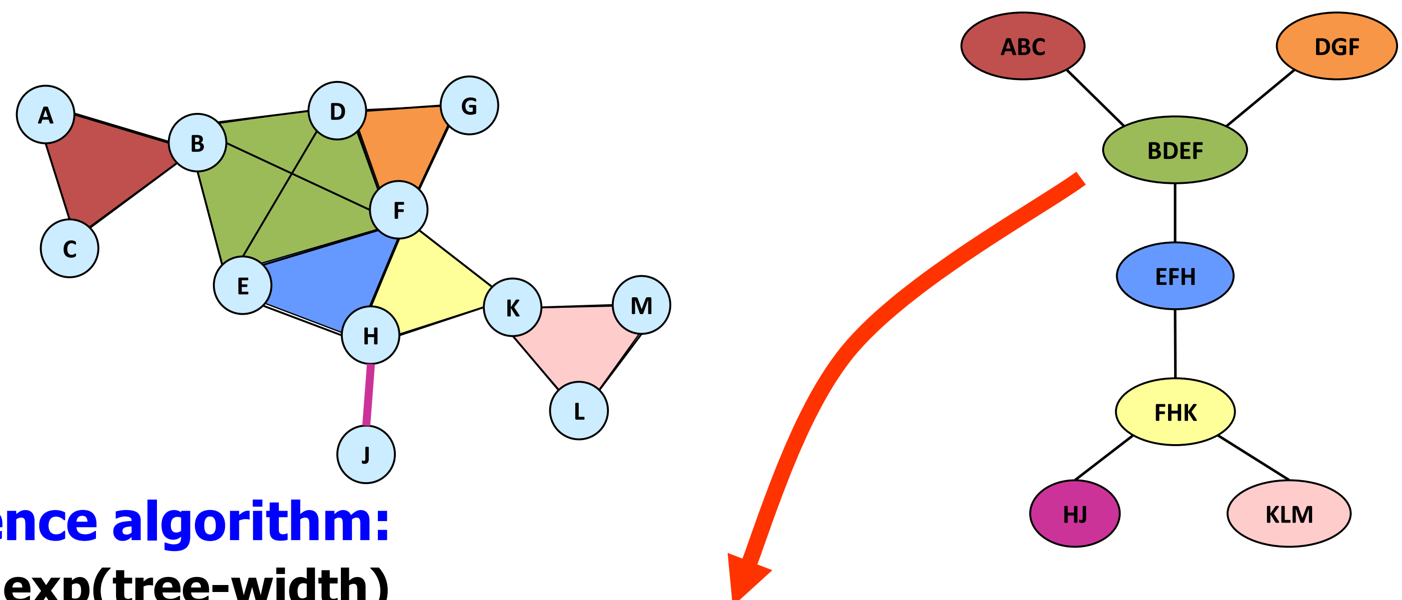
Trees are processed in linear time and memory

Transforming into a Tree

- **By Inference (thinking)**
  - Transform into a single, equivalent tree of sub-problems

- **By Conditioning (guessing)**
  - Transform into many tree-like sub-problems.

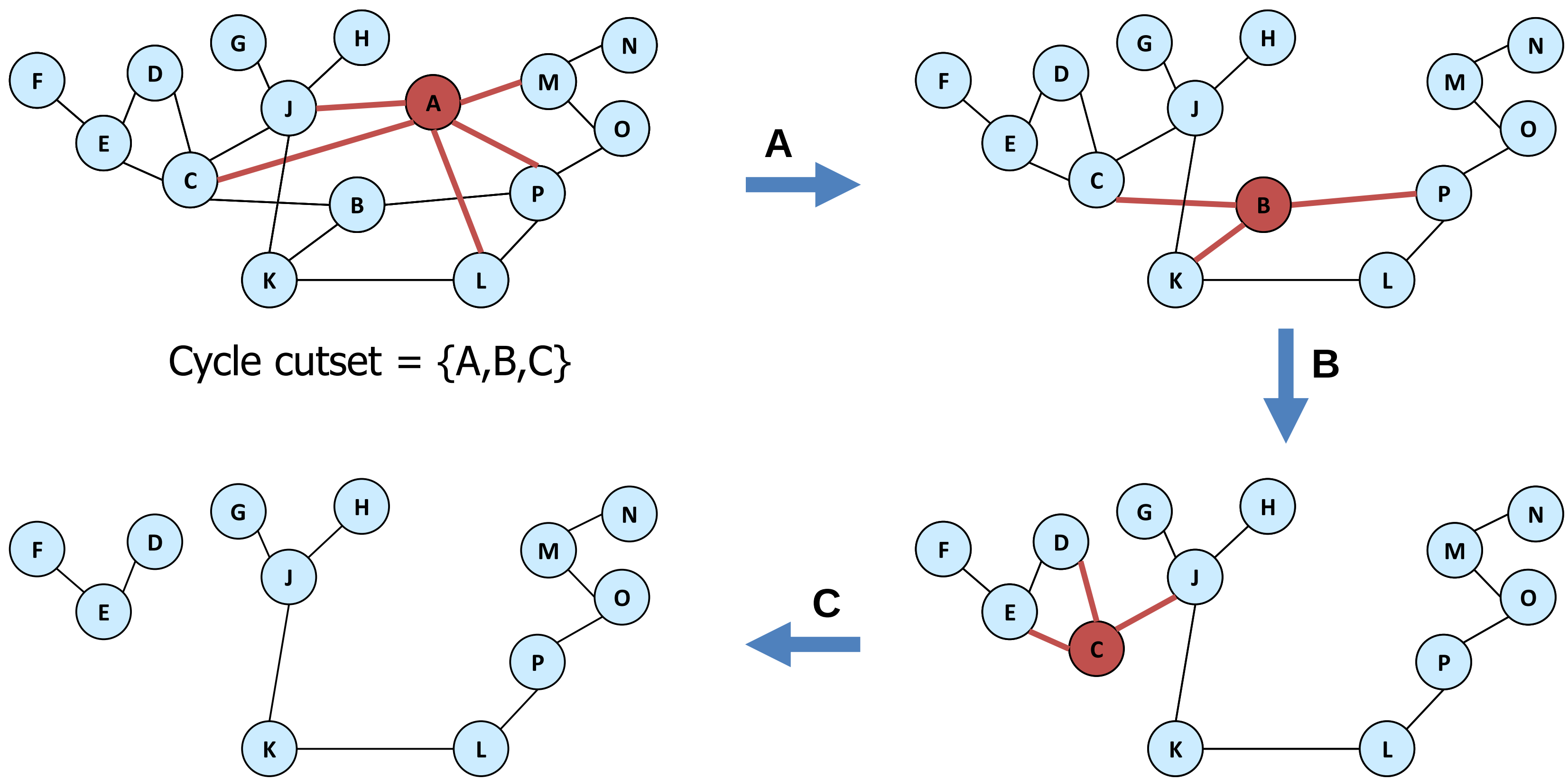
Inference and Treewidth



Inference algorithm:
Time: exp(tree-width)
Space: exp(tree-width)

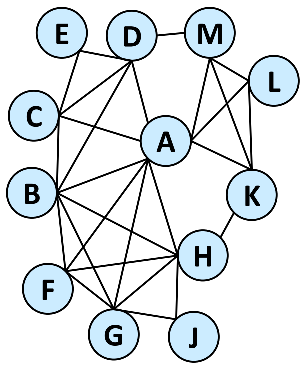
treewidth = 4 - 1 = 3
treewidth = (maximum cluster size) - 1

Conditioning and Cycle cutset

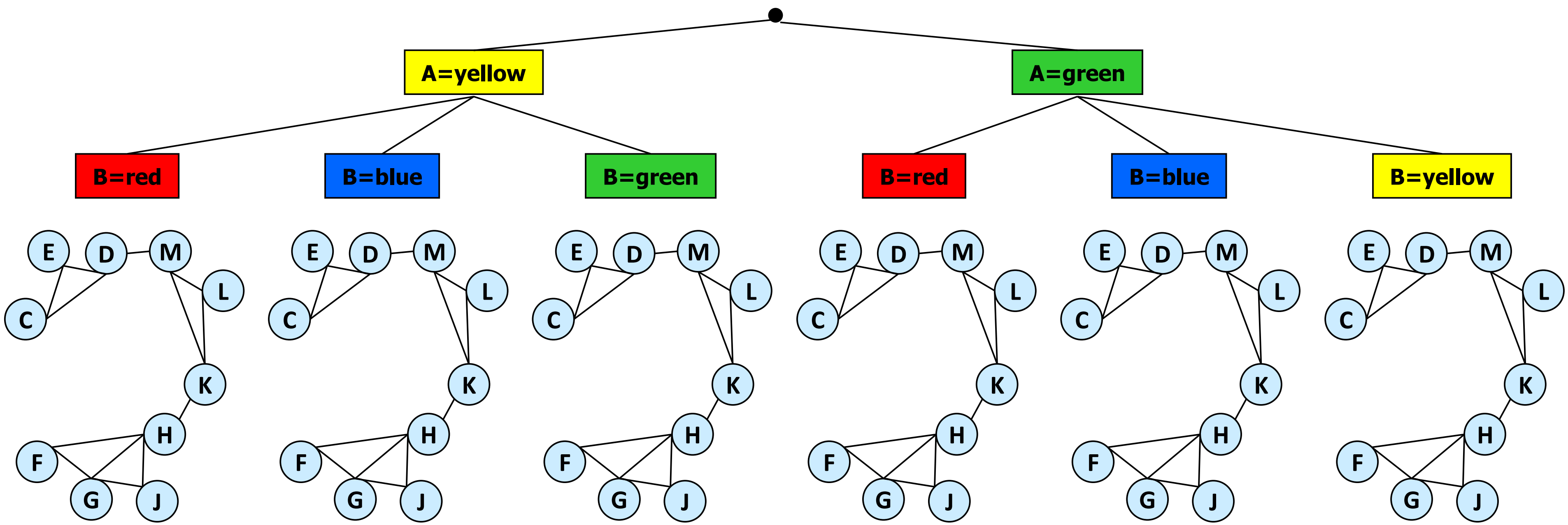


Search over the Cutset

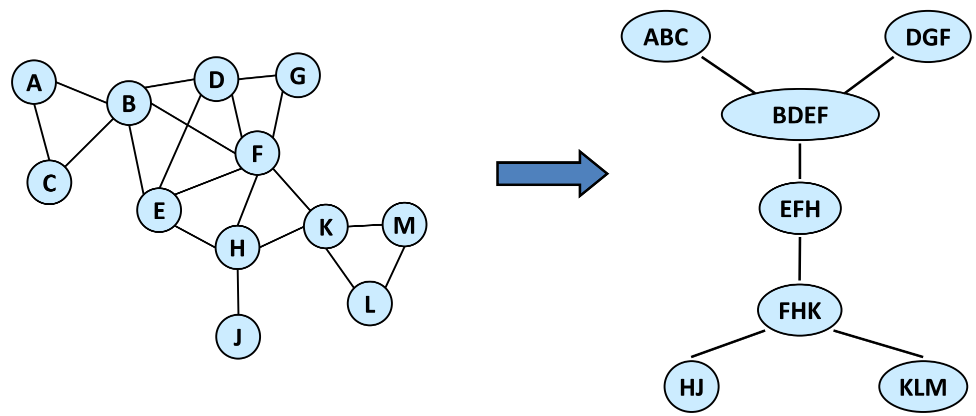
Graph Coloring problem



- Inference may require too much memory
- **Condition** on some of the variables

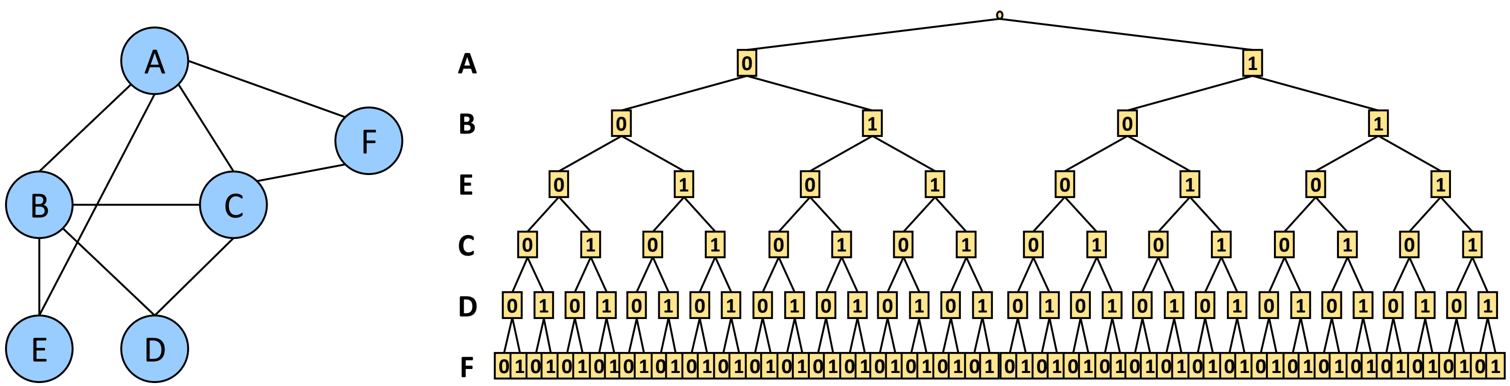


Bird's-eye View of Exact Algorithms



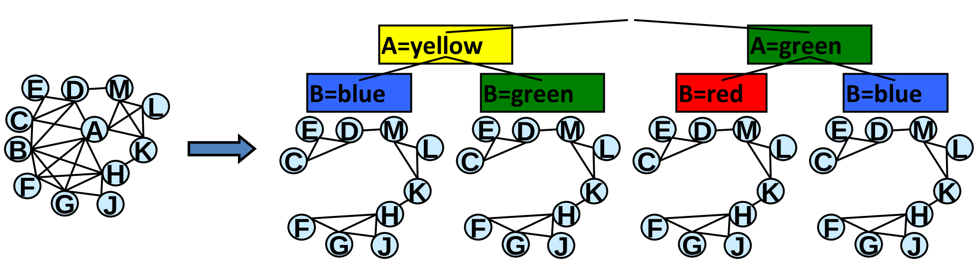
Inference

exp(w*) time/space



Search

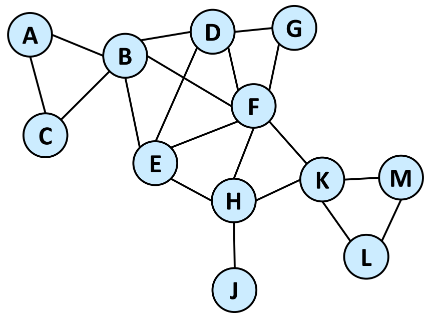
Exp(w*) time
O(w*) space

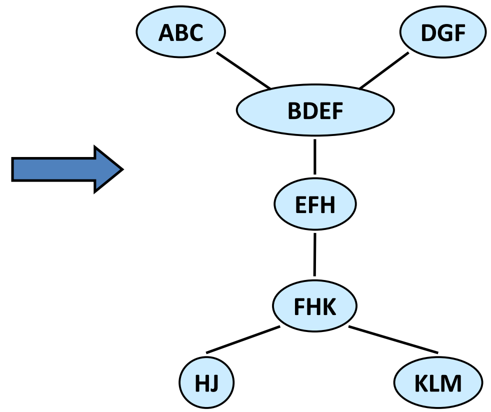


Search+inference:
Space: exp(q)
Time: exp(q+c(q))

q: user controlled

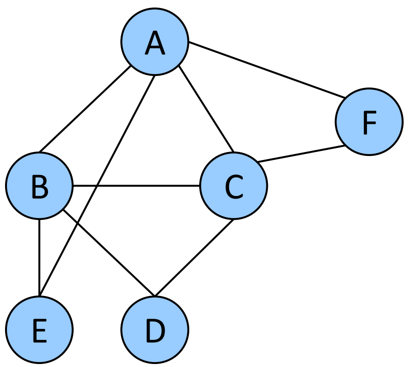
Bird's-eye View of Exact Algorithms

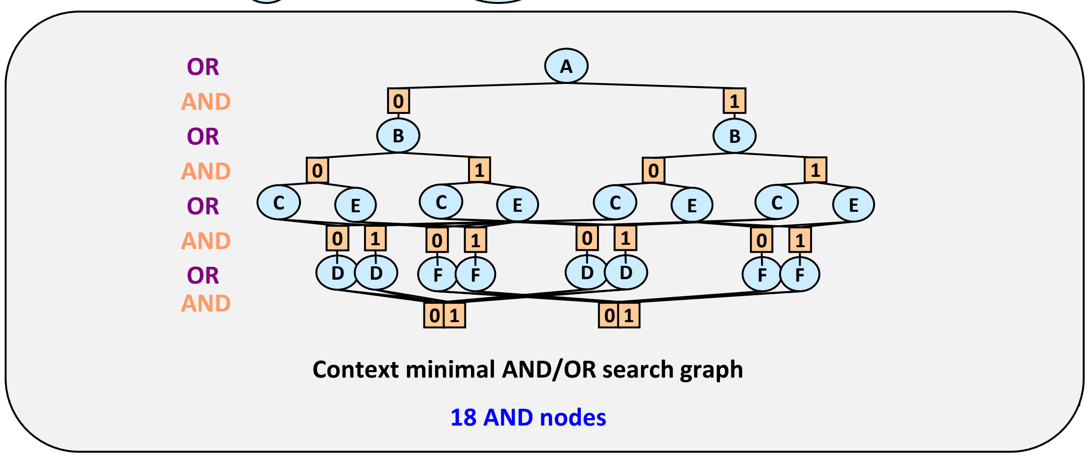




Inference

exp(w*) time/space

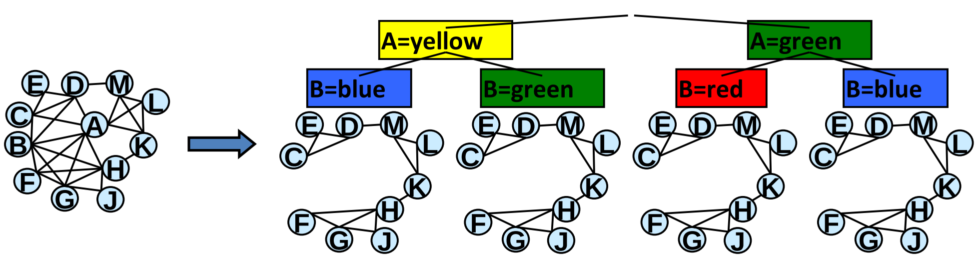




Search

Exp(w*) time

O(w*) space



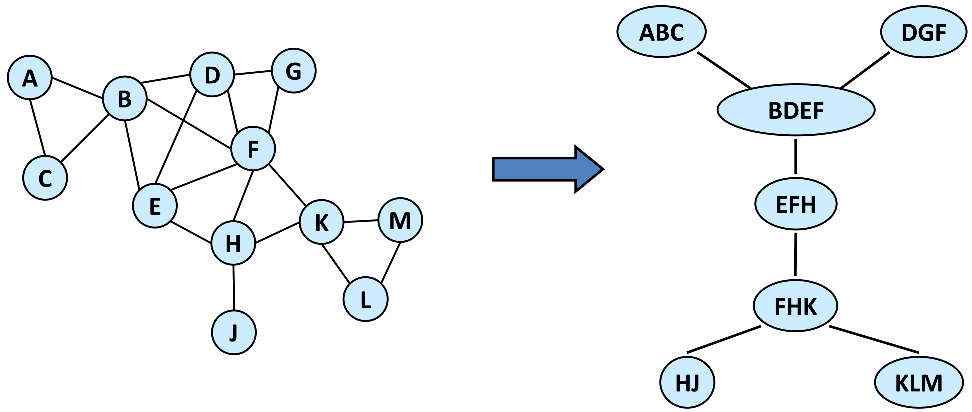
Search+inference:

Space: exp(q)

Time: exp(q+c(q))

q: user controlled

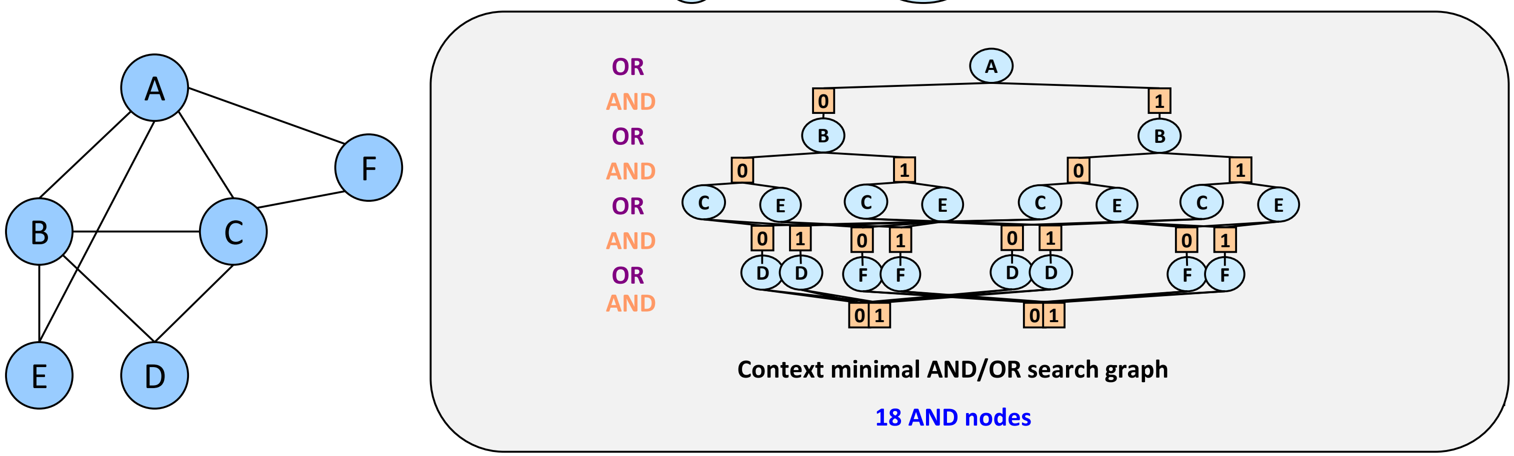
Bird's-eye View of Approximate Algorithms



Inference

↓

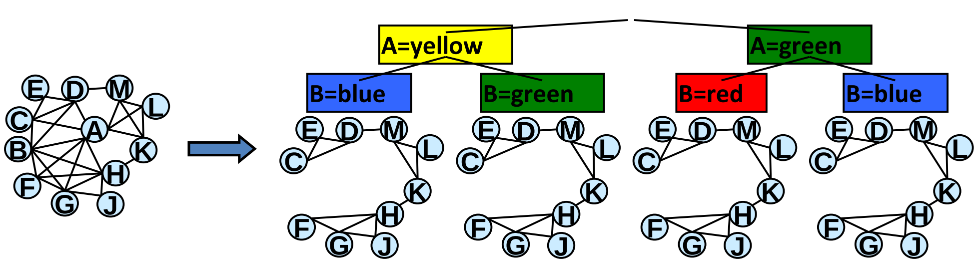
Bounded Inference



Search

↓

Sampling



Search + inference:

↓

Sampling + bounded inference